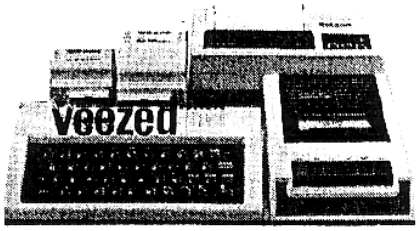
MICRO MAGIC



Beginner's Guide to the veezed ~ 1

AUTO

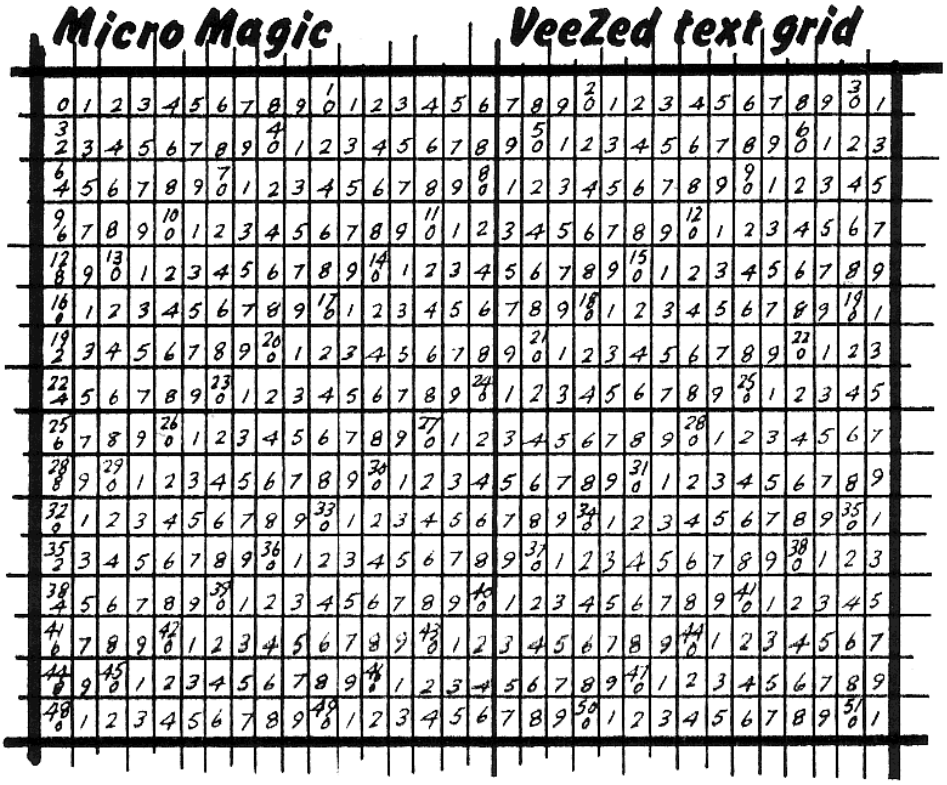
■ This, the first of 6-issues of MICRO MAGIC begins by introducing a method of AUTOMATIC LINE NUMBERING. In a future issue we will be publishing a feature telling you how to use the very many extra commands that are built in to your computer. The system of using AUTO described here, is similar to that used to acess all the others.

Begin by typing Line 10 in the following manner...

```
10  #100,10[RETURN]
```

Imagine that the HASH SIGN # is the word AUTO.
Now type `POKE31469,183:RUN` RUN is part of this routine.
As your line 10 is RUN the HASH SIGN has disappeared. So has 100,10.
All you will see is a line number. The command AUTO is in place and line numbers will be generated starting at 100 and will appear like magic in tens everytime you hit RETURN.
Do not add anything to line 10 for fear of overtyping the invisible AUTO 100,10. Should you use BREAK the Automatic Line Numbering will cease. To get them going again you need to type...
`POKE30945,175` and AUTO will resume from where it was halted.

Some of the instructions contained in MICRO MAGIC concerning AUTO were written before this easier way was discovered. We advise you stick with this method given here.
When the program is completed line 10 will need to be cancelled.



■ Gordon Browell, 13 Brookes Street, BIGGENDEN. Qld. 4621. ■

■ When your Granny was a girl pencil and paper games were very popular. NOUGHTS & CROSSES, BATTLESHIPS & SUBMARINES and more. This, our first MICRO MAGIC program, called BUTCHER was quite a popular pencil and paper game. Sometimes called HANGMAN the object of the game was to guess a secret word letter by letter while for every wrong guess part of the poor old butcher was drawn hanging from the gallows! The player has to guess the word before the ten parts of the whole man has been drawn. The word is represented by a cross for each letter and a clue to the word is given.

■ NOTE: All words and letters underlined should be entered in INVERSE.

BUTCHER

```
100 CLS:CLEAR:DIMB$(12):COLOR3,1:GOSUB620
```

Clear the screen. CLEAR will empty all variables that may have been used in a previous running of this or another program. DIMB$(12) reserves twelve subscripted B$ string variables. In a LOOP B$(1) to B$(12) will be used to store the individual letter of any word containing up to twelve letters. A colour is chosen and then we go to the sub-routine at line 620 to pick up the opening TITLE of this program. We will enter this sub-routine next.

```
620 SOUND1,5:CLS:PRINT"■■■
```

If you are using AUTO enter RUN. Then, for the first INPUT type in 160, and for INCR type in 10. The temporary line 160 will then appear. Type the true line 160 over the top of the old one.

160 A=LEN(A$):PRINT@257,"▀▀LETTERS▀▀";:PRINTUSING"##";A:COLOR5
"A" now contains the number of letters in the GUESSWORD. "LETTERS" has 2 SHIFTJ graphics at either end to add colour. PRINTUSING"##" ensures that no matter whether the GUESSWORD has one digit or two, the number printed will be in line with other numbers soon to be printed on the screen. "A", after the semicolon, prints the number of letters in the GUESSWORD just after the word LETTERS. Then change colour ready for the next line.

170 B=A+6:PRINT@289,"▀▀GUESSES▀▀":PRINT@321,"▀▀ALLOWED▀▀"B
"B" becomes the number of letters in the word plus 6. After printing GUESSES ALLOWED "B" is printed. This means that the player will have six more guesses than there are letters in the GUESSWORD. You may decide to change this after playing the game a few times. Both GUESSES and ALLOWED have 2 SHIFTJs either end.

180 PRINT@353,"▀▀ERRORS▀▀":PRINT@385,"▀▀ALLOWED▀▀ 10"
The words ERRORS and ALLOWED both have 2 SHIFTJs either end. As there are ten parts to the "HANGING" the errors permitted have been set at 10.

190 PRINT@40,Z$
PRINT the CLUE.

200 PRINT@161,"GUESS COUNT":PRINT@193,"ERROR COUNT"
The words GUESS COUNT and ERROR COUNT because of their lengths have no SHIFTJ graphics. Both will have numbers added after them as the game is played.

210 FORX=6TOA*2STEP2:PRINT@X,"X":NEXT
This LOOP is from screen position 6 to twice the length of the GUESSWORD. STEP2 will print an INVERSE "X" for each letter, two spaces apart.
And this might be a good time to test the program and get everything right before going any further. It is also a good idea to CSAVE what you have already typed to guard against losing the lot through a loss of power or some other catastrophe. But before you TEST RUN or CSAVE add a Temporary Line to prevent the program going through to the sub-routine at lines 620 to 640 after 210.

220 GOTO220

220 FORI=1TOA:B$(I)=MID$(A$,I,1):NEXT
FORI=1TOA is a LOOP where each letter of the hidden word is allocated a STRING VARIABLE in turn, as the LOOP progresses. B$(I) becomes B$(1) first and is made the first letter of A$ (which is the hidden word). MID$ selects "I" letter from A$, the "1" ensuring it selects only one letter. As the LOOP progresses B$(2) becomes MID$(A$,2,1) which is the second letter of the hidden WORD. The LOOP continues for the length of the WORD (from 1 to however many letters there are in the word).

230 L=448
The LOOP for counting the guesses and dealing with each one begins on the next line. It was necessary to make "L" 448 outside the playing loop. It is to be used for printing all the letters used in guesses at the bottom of the screen. Inside the game loop will be a command to increase "L" for each guess.

240 FORJ=1TOB:PRINT@70,"";:INPUT"GUESS LETTER";X$
As "B" is the length of the WORD plus 6 then this LOOP is going to run until the limit of guesses has expired. To put the INPUT in the right place on the screen we PRINTed at 70, nothing ("""), and by using a semicolon we ensured that the INPUT would be in the right place. So X$ becomes the first GUESSLETTER.

250 PRINT@L,X$:L=L+2
Print the letter (X$) at 448 then increase L by 2 so that the next GUESSLETTER will be printed two spaces from this one.

260 P=6:FORI=1TOA
Here we go with another LOOP! As this LOOP is INSIDE the game loop, beginning at line 240, this is what is called a NESTED LOOP. This one is for printing CORRECT guesses on top of those INVERSE Xs at the top of the screen. We start by making P equal 6, which is the location of the first X. The LOOP is to run for the length of the chosen word.

270 IFX$=B$(I)THENPRINT@P,X$
280 IFX$=B$(I)THENC=1
290 IFX$=B$(I)THENR=R+1
Here we have three IFs to pick out and deal with correct guesses. Line 270 says: if X$ is equal to B$(I),which is firstly B$(1) and contains the first letter of the chosen word then print at 6 the correctly guessed letter.
If the GUESSLETTER does not equal the first letter of the word then the computer ignores this line. When the LOOP comes round again it will check the GUESSLETTER with the second letter of the WORD, Z$(2), and so on, "P" will have to be increased each time so that correct guesses are printed in the right places.
Line 280 again tests the GUESSLETTER against each letter in turn of the chosen word. But this time it makes the variable C a 1 if the guess is correct.
NOTE: C is never more than 1. Even if the GUESSLETTER is in the hidden word more than once, C remains at 1. Only if the GUESSLETTER is WRONG then C will be 0.
Line 290 is similar to line 280 except that R is increased by 1 for every correct guess.

300 IFR=ATHEN400
When R (in line 290) reaches the same number as there are letters in the word then we know that the player has guessed the whole word. If the word, for instance, had been MISSISSIPPI, although there are 11 letters, R would have reached 11 in 4 guesses if all the guesses were correct. With a CLUE like "A LONG RIVER" it would be quite likely. So, when R does reach the same number as there are letters in the word then we must jump out of the game loop and end the game. This is where line 400 comes in.
(If using AUTO - BREAK - RUN - enter 400 then 10).

400 CLS:PRINT@197,"YOU'VE GOT IT!"A$:GOTO610
Clear the screen - print the message - then go to line 610.
(If using AUTO - BREAK - RUN - enter 610 then 10).

610 FORI=1TO3000:NEXT:RUN
This is just a pause to allow the player to read the message and RUN starts a new game.
Now back to line 310.
(If using AUTO - BREAK - RUN - enter 310 then 10).

310 P=P+2:NEXT
P (which is the screen positon of the INVERSE Xs) is now increased by 2 ready for printing the next guess at 8 if correct. NEXT is the end of the NESTED LOOP which was for checking GUESSLETTERS against each letter of the word. The NEXT for the Game LOOP is yet to come. We still have to deal with CORRECT and WRONG guesses. We haven't hung anybody yet!

```
320 IFC=0THEND=D+1
330 IFC=1THENPRINT@225,"▒▒▒ RIGHT!▒▒▒"ELSEPRINT@225,"▒▒▒ WRONG!▒▒▒"
340 IFC=0THENGOSUB410
```

Another three IFs. This time to make use of C.
Line 320 "IF C=0" means: if the GUESSLETTER was WRONG, (remember, C only becomes a 1 if the guess is CORRECT and stays 0 if the guess was WRONG), then D equals itself plus 1. We are counting the number of correct guesses. Even if a GUESSLETTER appears more than once in the word it is still counted as one correct guess. We will be using C shortly.
Line 330 says: if the guess was CORRECT then print "RIGHT!" but if the guess was WRONG then print "WRONG!".
NOTE: The words "RIGHT!" and "WRONG!" both have 2 SHIFTJ graphics at either end, and are in INVERSE starting with an INVERSE space.
Line 340 says: if the guess was WRONG then go to the sub-routine at 410 and come back on the command RETURN.

We are not going to type in the line 410 sub-routine yet as, from line 420 onwards, are all the HANGING graphics, and it will be easier to do all these together. They are the GALLOWS, HEAD, BODY and so on. We only need about five more lines to finish the actual playing of the game. Then we can TEST RUN and cure any BUGS before dealing with the hanging of the BUTCHER.

```
350 C=0:PRINT@205,D
```

Here we return C to zero ready for the next guess, and D (which is?) - yes, that's right - D is the number of correct guesses, is printed on the screen at position 205.

```
360 PRINT@173,J
```

J (which is the game LOOP) is the total number of guesses or the number times the LOOP has cycled. Print J at 173. It will be updated as the game progresses.

```
370 NEXT
```

...we've finally arrived at the end of the game LOOP! Many versions of BASIC would insist on NEXTJ, but the VeeZed does not.
When the computer reaches this line it means that all the permitted guesses have been used up, and the player has not guessed the word. The program goes on to the next line, which is....

```
380 CLS:PRINT@225,"SORRY!":PRINT" YOU'VE RUN OUT OF GUESSES."
390 PRINT:PRINT" THE WORD WAS "A$:GOTO610
```

Line 380 clears the screen and prints the message.
Line 390 says what the WORD was, and then goes to 610, which is already typed in. It is the 3 second pause followed by RUN to start another game.

■ There's enough program now to TEST RUN a game. But, before you do that you will need 2 temporary lines. You cannot use 400 because that is the line that says "YOU'VE GOT IT!", so, why not just use 392.

```
392 GOTO392
410 RETURN
```

Line 410 takes care of the GOSUB410 in line 340.

In many programs where POKEs are used, the TEST RUNNING of a program can cause a computer to CRASH! An error can send the computer into hysterics where the only cure is to switch off and on again. There is very little chance of this program CRASHING but it is still a good idea to CSAVE all that typing.

With the temporary lines in place RUN the program. If you are still using AUTO then type RUN100.
When you have corrected any errors and the program is running OK CSAVE it again. Now all that remains is to add all the PENALTIES for wrong guesses.

Ready to start again...

Delete Line 392.
Start AUTO at 410 - INCR 10.

In the next three lines we are going to use the SHORTFORM for "THEN". Instead of the command THEN we type a COMMA. This is exactly the same as THEN but takes less room in a program line.

```
410 IFD=1,440ELSEIFD=2,480ELSEIFD=3,490ELSEIFD=4,510
420 IFD=5,520ELSEIFD=6,530ELSEIFD=7,540ELSEIFD=8,550
430 IFD=9,560ELSEIFD=10,570
```

Notice that the computer must have an IF for each of the statements that are linked together with ELSEs. Now you know why the keyboard has THEN, ELSE and IF close together. While we are using a SHORTFORM for THEN, it can be mentioned that there is a SHORTFORM for PRINT. It is a QUESTION MARK. Strangely, when ? is used for PRINT it appears, when the program is LISTed, as the word PRINT. It can be used to squeeze a few extra characters into a line.

Lines 410 to 430 are saying: if the number of WRONG guesses is 1 then go to 440, but if the number of WRONG guesses are 2 then go to 480, etc. etc. The first WRONG guess will print the GALLOWS, the second WRONG guess will print the man's HEAD, and so on.
Remember that these three lines are part of a GOSUB and a RETURN will send the program back to the command following the GOSUB. The line containing the GOSUB is line 340, which is INSIDE the game loop.

Following are the lines that actually draw the HANGING man. There are ten parts and each part ends with a RETURN. To let you know what graphic characters to use we have added a code, (in brackets). Do not type these in. (12J) would simply mean... use 12 SHIFTJ graphic characters.

If you have been using AUTO then delete it now.

To delete AUTO simply type the AUTO line numbers one at a time each followed by RETURN.

AUTO line numbers are 0, 1 and 2.

As we proceed with the GRAPHICS,AUTO would interfere with the method of checking each part of your graphic picture as you go.

Each time you enter RETURN put an APOSTROPHE in front of the word, like 'RETURN.
This is a SHORTFORM of REM and will cause the computer to ignore the RETURN.
If you then enter the following Temporary Line: 599 GOTO599 you will be able to check each bit of graphics as they are entered by typing CLS:RUN.
After checking RUBOUT the APOSTROPHE before continuing.

Here we go..... Delete line 599

THE CODE IN BRACKETS AT THE END OF LINES IS TO LET YOU KNOW HOW MANY AND WHAT SORT OF GRAPHICS TO USE. (14J) MEANS USE 14 SHIFT J GRAPHICS). DO NOT TYPE THIS CODE.

```
440 COLOR8:PRINT@113,"▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀"        (14J)
450 FORI=157TO413STEP32
460 PRINT@I,"▀▀":NEXT                                  (2J)
470 PRINT@433,"▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀":RETURN          (1J+8T+5J)
480 COLOR5:PRINT@148,"▀▀▀":PRINT@180,"▀▄▄▀":RETURN  (1Q+1J+1W & 1Y+1J+1Y)
490 COLOR5:PRINT@213,"▄":COLOR6:PRINT@243,"▀▀▀▀▀▀"   (1Y & 2H+1S+2H)
500 PRINT@275,"▀▀▀▀▀▀":RETURN                          (5H)
510 COLOR6:PRINT@248,"▀▀":COLOR5:PRINT@250,"▀▀"   (2H & 1I+Q)
520 COLOR5:PRINT@239,"▀▀":COLOR6:PRINT@241,"▀▀":RETURN  (Q+1U & 2H)
530 PRINT@308,"▀▀▀":COLOR7:PRINT@340,"▀▀▀▀":RETURN  (3D & 3J)
540 PRINT@372,"▀":PRINT@404,"▀":RETURN                 (1J & 1J)
550 PRINT@374,"▀":PRINT@406,"▀":RETURN                 (1J & 1J)
560 COLOR4:PRINT@434,"▄▄":RETURN                       (1J & 1Y)
570 PRINT@433,"          "                             (10 SPACES)
580 PRINT@434,"▀▀▄ ▄▀▀":SOUND1,6                       (2J+1Y+SPACE+1Y+1J)
```

There are now only two more line to enter.
Line 590 will clear a space on the screen in which to print the secret word.

```
590 N=129:FORI=1TO10:PRINT@N," ":N=N+1:NEXT:PRINT@129,A$
```

N is the screen location for the LOOP to print nothing! By nothing we mean empty quotes, " " and as N is increased by 1 each time the next "nothing" to be printed will be one space further on each time. When ten spaces have been cleared, we then print in that space the secret word.

```
600 PRINT@151,"EE-EEK!":SOUND31,9
```

The program is now complete.

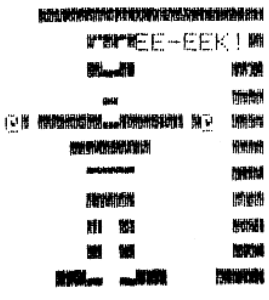
As you will appreciate, this MICRO MAGIC program has been checked for correctness. However, experience tells us, that the VeeZed Gremlin may have hidden an error which escaped detection. We apologise if this is the case and hope that you will be able to put it right and let us know what was wrong.

You may have read that programs should be STRUCTURED. Well, BUTCHER was not structured in the manner that many programmers would recommend. When writing a program from scratch it is usually necessary to draw up a FLOW CHART. A FLOW CHART is something that looks like a tree. The main stem has a lot of GOSUB branches. To design a program one draws a vertical line to represent the program from the first line to the end. This line would be broken by inserting boxes in which you write words such as "TITLE", "INPUT(WORD)", "INPUT (CLUE)" and so on. Branches leading to other boxes would then radiate from the main stem boxes where sub-routines are noted with such words as "FIND LENGTH OF WORD", "PUT IN ERROR TRAP", and anything else that is needed before RETURNing to the main stem again. You will probably use BASIC as a form of shorthand when drawing up a flow chart. Once the FLOW CHART looks as though you've covered everything, you can then add consecutive numbers to the boxes on the main stem, and larger block numbers to all the sub-routine boxes. The idea of keeping the sub-routine line numbers higher than the main stem is to help speed up the running of the program. Any sub-routine which is not used more than once would have the highest block of line numbers. You see, every time a LOOP cycles, such as in BUTCHER, the computer scans every line from the lowest line number to the next one to be processed. Programs with heaps of DATA always need to have the DATA at the bottom of the program out of the way of LOOPS and GOSUBs. But, more of this later. MICRO MAGIC will be publishing programs where this can be demonstrated. In the mean time, you might try drawing up a FLOW CHART for BUTCHER. As a clue you should have the ERROR TRAP in line 140 as a sub-routine, which means the GOTO120 should still remain in the main stem of the program.

When we compare the GUESSLETTER with the first and subsequent letters of the WORD, this is possible because the computer only deals in numbers. Every letter has what is called an ASCII NUMBER. If you were to type PRINTASC("B") the computer would print 66. Later we see how by changing the ASCII numbers we can make a printer or plotter print lower-case letters. We can also compare letters and have them arranged in alphabetical order.

BUTCHER

```
100 CLS:CLEAR:COLOR3,1:GOSUB620
110 PRINT"[illegible]":SOUND28,1
120 COLOR7:PRINT@195,"ENTER WORD";:INPUTA$
130 PRINT@206,"[illegible]":SOUND28,3
140 IFLEN(A$)>10THENPRINT@234,"[illegible]":GOSUB620:GOTO120
150 PRINT@259,"ENTER CLUE";:INPUTZ$:CLS
160 A=LEN(A$):PRINT@257,"[illegible]";:PRINTUSING"##";A:COLOR5
170 B=A+6:PRINT@289,"[illegible]":PRINT@321,"[illegible]"B
180 PRINT@353,"[illegible]":PRINT@385,"[illegible] 10"
190 PRINT@40,Z$
200 PRINT@161,"[illegible]":PRINT@193,"[illegible]"
210 FORX=6TOA*2+4STEP2:PRINT@X,"[illegible]":NEXT
220 FORI=1TOA:B$(I)=MID$(A$,I,1):NEXT
230 L=448
240 FORJ=1TOB:PRINT@70,"";:INPUT"[illegible]";X$
250 PRINT@L,X$:L=L+2
260 P=6:FORI=1TOA
270 IFX$=B$(I)THENPRINT@P,X$
280 IFX$=B$(I)THENC=1
290 IFX$=B$(I)THENR=R+1
300 IFR=ATHEN400
310 P=P+2:NEXT
320 IFC=0THEND=D+1
330 IFC=1THENPRINT@225,"[illegible]"ELSEPRINT@225,"[illegible]"
340 IFC=0THENGOSUB410
350 C=0:PRINT@205,D
360 PRINT@173,J
370 NEXT
380 CLS:PRINT@225,"SORRY!":PRINT" YOU'VE RUN OUT OF GUESSES."
390 PRINT:PRINT" THE WORD WAS "A$ GOTO610
400 CLS:PRINT@197,"YOU GOT IT!  "A$:GOTO610
410 IFD=1,440ELSEIFD=2,480ELSEIFD=3,490ELSEIFD=4,510
420 IFD=5,520ELSEIFD=6,530ELSEIFD=7,540ELSEIFD=8,550
430 IFD=9,560ELSEIFD=10,570
440 COLOR8:PRINT@113,"[illegible]"
450 PRINT@113,"[illegible]"'   DELETE THIS DUPLICATE LINE
460 FORI=157TO413STEP32:PRINT@I,"[illegible]":NEXT
470 PRINT@433,"[illegible]":RETURN
480 COLOR5:PRINT@148,"[illegible]":PRINT@180,"[illegible]":RETURN
490 COLOR5:PRINT@213,"[illegible]":COLOR8:PRINT@243,"[illegible]"
500 PRINT@275,"[illegible]":RETURN
510 COLOR8:PRINT@248,"[illegible]":COLOR5:PRINT@250,"[illegible]":RETURN
520 COLOR5:PRINT@239,"[illegible]":COLOR8:PRINT@241,"[illegible]":RETURN
530 PRINT@308,"[illegible]":PRINT@340,"[illegible]":RETURN
540 PRINT@372,"[illegible]":PRINT@404,"[illegible]":RETURN
550 PRINT@374,"[illegible]":PRINT@406,"[illegible]":RETURN
560 COLOR4:PRINT@434,"[illegible]":RETURN
570 PRINT@433,"                "
580 PRINT@433,"                ":PRINT@434,"[illegible]":SOUND1,6
590 N=129:FORI=1TO10:PRINT@N," ":N=N+1:NEXT:PRINT@130,A$
600 PRINT@151,"EE-EEK!":SOUND31,9
610 FORI=1TO3000:NEXT:RUN
620 SOUND1,5:CLS:PRINT"[illegible]"
630 PRINT@32,"[illegible]"
640 PRINT@64,"[illegible]":RETURN
```

[illegible] THIS LIST MAY HELP OVER COME PROBLEMS AND WILL ALLOW YOU TO READ THE PROGRAM. PLEASE DELETE THE DUPLICATED COMMAND IN LINE 580 AND CHANGE LINE 570 TO READ...

```
PRINT@70,
```